Sinclair ZX SPECTRUM Userclub

Clubinfo Nr. 3 (96) vom 01. März 1992

Rolf Knorre - Postfach 20 01 02 - 5600 Wuppertal 2
Telefon 0202 / 55 66 89 - Telefax 0202 / 59 47 21
Stadtsparkasse Wuppertal (BLZ 33050000) Kto. 853382

Hallo Freunde,

eigentlich sollte das Info diesen Monat auch wieder dicker werden. Leider habe ich von Euch nicht genug Material dazu erhalten. Die Mitarbeit am Info hat deutlich nachgelassen. Ist das der allgemeine Winterschlaf? Rafft Euch auf! Nächsten Monat würde ich gerne wieder rund 30 Seiten rausschicken. Vielen Dank.

Viel Spaß mit dem neuen Info.

Rolf Knorre

Wilhelm Wlecke, Lönsweg 92, 4904 Enger Tel. 05224/3449

Folge 25

Enger, den 10.2.92

Hallo Spectrum-User!

Auch diesmal muß ich, wie schon öfter, mit einer Korrektur meines letzten Beitrags beginnen! In der Tabelle "Organisation des Bildschirmspeichers beim ZX-SPECTRUM" ist mir ein kleiner Fehler unterlaufen: Bildzeile 8 beginnt natürlich bei Adr. 18432 ($4800) und nicht wie dargestellt bei 18400 (bei dieser Adr. beginnt nämlich die letzte Pixelreihe der 7. Bildzeile)!

Nun aber endlich zum angekündigten Assemblerlisting der schon mehrfach behandelten MC-Routine "Hardcopy". Da sicherlich nicht jeder MC-Interessierte selbst einen Assembler besitzt und dadurch dessen Anweisungen kennt, an dieser Stelle auch mal ein paar Erläuterungen zu den immer wieder vorkommenden Begriffen.

Die Startadr. wird in Zeile 120 mit ORG 65050 festgelegt (ORG = Origin = Anfang). In den Zeilen 170-410 werden vom Programm benutzten Adressen und Variablen mit Namen (=Labels) versehen. Dazu wird die Anweisung EQU (equal = gleich) benutzt. Labels vereinfachen das Schreiben von Assemblerprogrammen ungemein, da oft die spätere Adresse von Sprungzielen oder Unterprogrammen innerhalb des Programms noch nicht bekannt sind. Außerdem kann ein Programm so durch Ändern von 'ORG' und erneutes Assemblieren leicht für eine neue Startadresse konfiguriert werden und daneben kann das Label durch geschickte Namenswahl die Funktion des Unterprogramms, der Variablen usw. kenntlich machen.

Wie noch bekannt sein dürfte, wird "Hardcopy" mit dem Aufruf:

PRINT USR startadresse,<format>,<1.zeile>,<zeilenanzahl>,<1. spalte>,<spaltenanzahl>

gestartet. Die in <...> gesetzten Parameter sind optional und müssen durch Kommas oder Semikolons getrennt werden, wenn man sie benutzt. Die für 'format' zulässigen Werte sind:

0 (00000000) =norm. Größe		128 (10000000) =norm. Größe, invert.	
1 (00000001) =dopp. Breite		129 (10000001) =dopp. Breite, inver.	
2 (00000010) =dopp. Höhe		130 (10000010) =dopp. Höhe, invert.	
3 (00000011) =dopp. Größe		131 (10000011) =dopp. Größe, invert.	

Folgende Werte sind bei den übrigen Parametern erlaubt:

1.zeile	= 0 bis 23	zeilenanzahl	= 1 bis 24
1.spalte	= 0 bis 31	spaltenanzahl	= 1 bis 32

Der eigentliche Anfang (hier Adr. 65050 = $FE1A) der Routine ist in Zeile 470. Dort wird sofort zum Unterprogramm 'STEST' verzweigt. Dies prüft, ob dem Aufruf <PRINT USR startadr> ein Komma oder Semikolon folgt. Damit wird festgestellt, ob die besagten Parameter folgen und ausgewertet werden müssen. Wurde keins von beiden gefunden, wird als Druckformat 'Normale Größe' angenommen und entsprechend verzweigt (Zeile 490). Sonst wird der Parameterwert mit Hilfe von ROM-Routinen aus der BASIC-Zeile geholt (RST $20), geprüft ob numerisch (oder String) und auf dem Kalkulatorstapel abgelegt (Zeile 540). Die nächste ROM-Routine holt die eben abgelegte Integerzahl ins A-Register, mit AND $83 werden alle zur Zeit nicht interessierenden Bits ausgeblendet und das Ergebnis ist das Druckformat. Dies landet nun in der Adresse der Variablen 'FORMFLG' (Zeile 580)!

Als nächstes werden die Standardwerte für Zeilen-/ Spaltenanzahl festgelegt, bevor auf weitere Paramater geprüft wird. Sind keine da, bleibt's dabei, sonst bringt eine ROM- Routine die beiden nächsten durch Komma getrennten Zahlen auf den Kalk.-Stapel. Die 2. Zahl (=zuletzt abgelegte) wird gleich zurückgeholt (Zeile 690) ins A-Reg. und als Zeilenanzahl an der entsprechende Variablenadr. abgelegt. Falls es 0 war (wird mit 'AND A' ermittelt), heißt daß, es ist kein Zeichen zu drucken. Daher wird 'ENDE' aufgerufen und ins BASIC zurückgekehrt. Sonst wird auch die 1. Zahl ins A-Reg. geholt (= Zeilennummer) und auf zulässige Größe abgecheckt. Bei falschem Wert gibt's eine Fehlermeldung mit Programmabbruch, andernfalls wird das gleiche Spiel mit den Parametern '1. Spalte, Spaltenanzahl' wiederholt.

Die Parameterübergabe mittels PRINT USR-Aufruf ist sehr elegant gelöst. Man könnte jedoch auf die Auswertungsroutinen verzichten, wenn die Parameter schon durch BASIC-Abfragen auf Zulässigkeit geprüft wurden. Dann brauchen die Werte nur noch mit POKE xxxxx,yyy in die jeweiligen Variablen gebracht werden und die Hardcopy-Routine könnte noch um einiges kürzer werden.

Der wichtigste Programmteil beginnt in Zeile 1100. Erst wird der Druckerkanal geöffnet mit Hilfe einer ROM-Routine (hier Kanal 3), gefolgt von der ESC-Sequenz zur Zeilenabstandsänderung auf 8/72 Inch damit die Bildzeilen lückenlos aneinander anschließen.

Dann muß der Grafikmodus aktiviert werden. Dazu wird bei den meisten EPSON-kompatiblen Drucker die Codefolge 27,75,n (für Grafikdruck in normaler Dichte benutzt). In BASIC würde der entsprechende Befehl in etwa so aussehen:

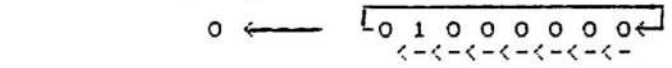
```
LPRINT CHR$ 27; "K"; CHR$ n;
```

Was hat es mit dem Grafikmodus überhaupt auf sich? Bekanntlich besteht eine komplette SPECTRUM-Zeile aus 32 Zeichen, jedes Zeichen wieder aus 8 Pixelspalten bei Ausgabe an den Drucker (siehe letztes Info). Das macht für normal breite Hardcopies des SPECTRUM-Screens also 8 * 32 = 256 Pixelspalten. Diese Anzahl entspricht dem Wert n und der (oder ein anderer errechneter z.B. bei Ausschnitten oder vergrößerten Bildern) muß dem Drucker unbedingt mitgeteilt werden, damit er weiß, wann eine Zeile zu Ende ist. Mit Line feed und Carriage return wie im Text- ist im Grafik-Modus nichts zu machen (erst wieder, wenn die Grafikzeile beendet und der Drucker sich automatisch in den Textmodus zurückgestellt hat, siehe Zeile 1930-1960) weil auch diese Steuercodes nur als Pixelmuster interpretiert werden. Bei meinem alten Drucker (SHINWA CP 80, 9 Nadeln) lassen sich im Normalbetrieb 80 Zeichen in einer Zeile unterbringen, d.h., im Grafikmodus sind 8*80 =640 Pixelspalten verfügbar (für anderen Druckern können natürlich andere Maximalwerte gelten)! Die max. Druckbreite von 640 Punkten entspricht beim CP 80-Drucker 7.5 Inch oder 190,5 mm, d.h. ein Punkt ist 7.5 / 640 = 0.0117 Inch oder ca. 0,298 mm breit. Die volle Breite wird bei meinem CP 80 von der Hardcopy-Routine allerdings nicht ganz ausgenutzt, da ein doppelt breites Bild nur 256 * 2 = 512 Punkte (512 * 0.0117 = 5.99 Inch oder 152,6 mm) breit werden kann. Kennt man also die bedruckbare Papierbreite und die maximale Punktezahl im entsprechenden Modus (siehe Druckerhandbuch), läßt sich die erzielbare Bildgröße leicht berechnen. Auf andere Grafikmodi (z.B. doppelte Dichte usw.) will ich hier nicht näher eingehen.

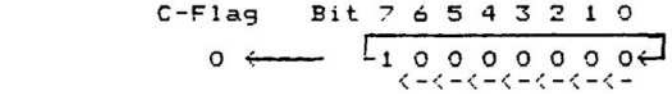
Interessant ist, wie Wert n im Programm errechnet wird (Zeile 1230-1360)! Dazu werden wieder Rotationsbefehle verwendet, nachdem die Spaltenanzahl ins A-Reg. übertragen wurde. Dazu ein Beispiel mit einer Spaltenzahl von 32 für ein ganzes Bild:

A-Registerinhalt: 32 = $20 = 00100000

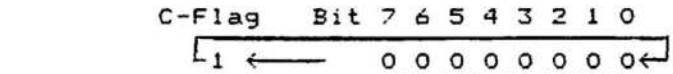
A-Register nach: Rotiere A-Reg. links kreisförmig = **RLCA** :



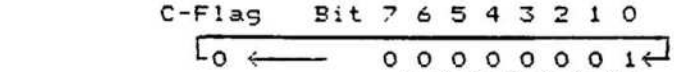
A-Register nach: Rotiere A-Reg. links kreisförmig = **RLCA** :



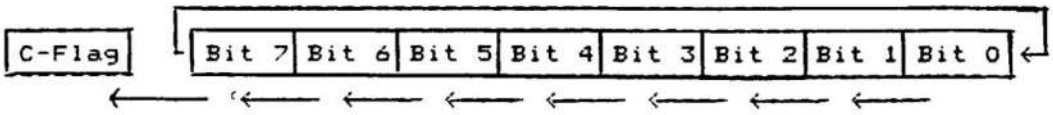
A-Register nach: Rotiere A-Reg. links durchs C-Flag = **RLA** :



H-Register nach: Rotiere H-Reg. links durchs C-Flag = **RL H** :
(H-Reg. wurde vorher gelöscht!)



Das Funktionsschema für RLCA (RLA und RL H siehe letztes Info):

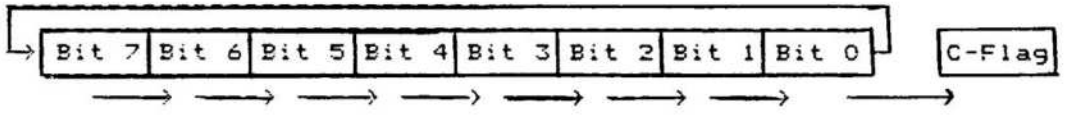


Wenn nach Prüfung von FORMFLG mit BIT 0,(IY+88) (IY+88 =Adr. 23698) festgestellt wird, daß keine X-Streckung vorgesehen ist, kann sofort bei NOSTR (Zeile 1380) der letzte Wert im A-Reg. als Low-Byte des Wertes n zum Drucker geschickt werden (im Beispiel 0). Als High-Byte wird anschließend der Wert aus H nach Übergabe ins A-Reg. hinterhergeschickt (Zeile 1390-1400), d.h., es wurde hier im Beispiel die 2-Byte-Zahl 00000001 00000000 = $0100 = 256 abgeliefert, also die benötigte Bildbreite in Pixeln!

Bevor die im letzten Info ausführlich erklärte Bildbyte-Umwandlung in druckfähigen Code vorgenommen werden kann (Zeile 1670-1830), muß erst an Hand der in Zeile 1410-1430 übergebenen Werte die Anfangsdr. des umzuwandelnden Bildbytes berechnet werden. Dazu dient das in Zeile 1470 aufgerufene Unterprogramm SCADR! Was dort abläuft, läßt sich am einfachsten durch Verfolgen der Registerinhalte in Binärdarstellung erkennen! Dazu 2 Beispiele mit unterschiedlichen Zeilen/Spalten-Nummern:

Beispiel 1:			**Beispiel 2:**		
ZLNR in H-Reg. = 8 = 00001000 SPNR in L-Reg. = 0 = 00000000			ZLNR in H-Reg. = 18 = 00010010 SPNR in L-Reg. = 10 = 00001010		
		C-Flag			C-Flag
LD A,H	A=$08 = 00001000		LD A,H	A=$12 = 00010010	
AND $07 = 7	00000111		AND $07 = 7	00000111	
Ergeb.:	A=$00 = 00000000		Ergeb.:	A=$02 = 00000010	
RRCA	A=$00 = 00000000	0	RRCA	A=$01 = 00000001	0
RRCA	A=$00 = 00000000	0	RRCA	A=$80 = 10000000	1
RRCA	A=$00 = 00000000	0	RRCA	A=$40 = 01000000	0
OR L	L=$00 = 00000000		OR L	L=$0A = 00001010	
Ergeb.:	A=$00 = 00000000		Ergeb.:	A=$4A = 01001010	
LD L,A	L=$00 = 00000000		LD L,A	L=$4A = 01001010	
LD A,H	A=$08 = 00001000		LD A,H	A=$18 = 00010010	
AND $18 = 24	= 00011000		AND $18 = 24	= 00011000	
Ergeb.:	A=$08 = 00001000		Ergeb.:	A=$10 = 00010000	
ADD $40 = 64	= 01000000		ADD $40 = 64	= 01000000	
Ergeb.:	A=$48 = 01001000		Ergeb.:	A=$50 = 01010000	
LD H,A	H=$48 = 01001000		LD H,A	H=$50 = 01010000	
In HL steht nun die Screen-Adresse $4800 = 18432!			In HL steht nun die Screen-Adresse $504A = 20554!		

Wieder wird ein Rotationsbefehl benutzt: RRCA, d.h. rotiere das A-Reg. rechts kreisförmig! Was man sich darunter vorzustellen hat, verdeutlicht hoffentlich das folgende Funktionsschema:



Wozu dient eigentlich das AND $07? Damit wird die jeweilige Zeilennr. in 8-er Blöcke zerlegt und nur die Information, welche Position die Zeile innerhalb dieses Blocks einnimmt (0-7) ausgewertet. Block 1 umfaßt die Zeilennr. 0-7, Block 2 Zeile 8-15 und Block 3 Zeile 16-23! Im Beispiel 1 ist Bildzeile 8 (ZLNR) die 0. Zeile im 2. Block. In Beispiel 2 hat ZLNR 18 dagegen die Position 2 im 3. Block.

Das 3-malige RRCA errechnet daraus die entsprechende Pixeladr. (Low-Byte) innerhalb des Blocks. Dazu ist die schon ermittelte Position der Zeile im Block nur mit 32 zu multiplizieren und genau das ist hier geschehen. Warum * 32? Wie noch aus dem letzten Info bekannt sein dürfte, liegen die Anfangsadressen der einzelnen Bildschirmzeilen im Speicher 32 Bytes auseinander!

OR L blendet ins Ergebnis der Berechnung anschließend die Spaltennr. (SPNR) ein und legt so die genaue Pixelpositon der zu druckenden Pixelspalte erst Blockunabhängig fest. Um auch das High-Byte der Adresse zu ermitteln, wird mit AND $18 (=24) der Rest der Zeilennr. ausgewertet. Es interessieren nur noch Bit 3-4. Wer sich die möglichen Zeilennummern (0-23) in Binärschreibweise ansieht, wird feststellen das mit AND $18 als Ergebnis nur 0, 8 oder 16 herauskommen kann, also jeweils die Zeilennr., an der ein neuer Block beginnt. Nachdem dazu $40 (das Highbyte der Bildspeicheranfangsadr. =$4000 =16384) addiert wurde und das Ergebnis ins H-Register verlagert wurde, haben wir unsere gesuchte Pixeladresse im HL-Register stehen.

Nun können also endlich die Bildbytes ausgelesen, in den Puffer SBUFF abgelegt und in druckbaren Code verwandelt werden? Ganz so schnell geht's leider nicht, denn es ist ja noch zu berücksichtigen, ob in X-, Y- oder beide Richtungen gestreckt werden soll! Am einfachsten ist nur X-Streckung (also von FORMFLG BIT 0 gesetzt). Dann wird der aufbereitete Code einfach 2 mal nacheinander zum Drucker geschickt (Zeile 1810-1860).

Wesentlich komplizierter ist der Fall bei Y-Streckung! Dafür ist es auch wieder erforderlich, jeden Bildpunkt zweimal auszugeben, nur diesmal untereinander. Das bedeutet, daß aus einer 8-Bit Pixelspalte plötzlich 16 Bit würden. Weil aber dem Drucker weiterhin nur ein 8 Bit großer Wert übergeben werden kann, muß das Drucken in 2 Durchläufen erfolgen. Dazu wird das erste Bildbyte aus der ermittelten Pixelzeilenadr. normal in SBUFF geladen (Zeile 1570-1580), die SBUFF-Adr. erhöht und mit BIT 1,(IY+88) geprüft, ob Y-Streckung erfolgen soll (Zeile 1590-1610). Wenn ja, wird das gleiche Byte noch mal in SBUFF geschrieben, bevor die SBUFF-Adr. erneut erhöht wird und die Pixelzeilenadr. auf die nächste Pixelzeile gesetzt wird. SBUFF ist also bei Y-Streckung voll, wenn die 4. Pixelzeile ausgelesen wurde. Dann muß die Codeumwandlung erfolgen und die Punktmuster der oberen Hälfte des Zeichens werden zum Drucker geschickt (Zeile 1670-1820). Das ganze Spiel wiederholt sich natürlich mit den benachbarten Bytes, bis die angegebene Spaltenzahl erreicht ist und der Ausdruck erfolgt. Mit Hilfe der Zeilen 1970-2030 wird nun festgestellt, ob erst die obere Hälfte oder schon die ganze Zeile bei Y-Streckung bearbeitet wurde. War erst der 1. Durchlauf dran, geht's mit JP Z,COPLN zurück (Zeile 2020) nach COPLN (Zeile 1190), ohne das ZLNR erhöht wurde (das geschieht erst ab Zeile 2050). So wird nochmal die gleiche Pixelzeilenadr. ermittelt wie im 1. Durchlauf, aber bei Zeile 1500 wird nun festgestellt, daß der 2. Durchlauf dran ist und deshalb das High-Byte der Pixelzeilenadr. um 4 erhöht werden muß (Zeile 1520-1550). Damit zeigt H auf das oberste Byte der unteren Zeichenhälfte und nun beginnt das beschriebene Spiel erneut.

Bei Streckung in X- und Y-Richtung werden beide Varianten abgearbeitet. Was geschieht aber bei invertiertem Ausdruck? Eine einfache Sache: ist BIT 7,(IY+88) gesetzt (=1), wird vor Übergabe des umgerechneten Codes an den Drucker noch der Befehl CPL aufgerufen. Der dreht alle Bits um, d.h aus z.B. 10011001 wird 01100110 und damit ist die Invertierung schon erreicht.

Bevor das Programm endet wird mit der ESC-Sequenz 27,10,12 der Zeilenabstand wieder auf den ursprünglichen Wert zurückgesetzt (Zeile 2080-2130). JP ENDE (Zeile 2140) führt ins BASIC zurück!

So, das war's mal wieder! Es gäbe sicherlich noch manches Detail in der Routine, mit dem man sich noch ausführlicher befassen könnte, aber es soll nun reichen. Hoffentlich ist alles halbwegs verständlich und fehlerfrei geworden. Bis demnächst!

U. Hahn

```
                0010 ;################################################################
                0020 ;#                                                              #
                0030 ;#  Hardcopy-Routine für Normalpapierdrucker (Nach 'CK' Heft 11/85)  #
                0040 ;#                                                              #
                0050 ;#  Filename: Quelltext: Copy-t                  Datum: 14.01.92  #
                0060 ;#                                                              #
                0070 ;# Kommentar: Wilhelm Wiecke, Lönsweg 92, 4904 Enger, Tel. 0224/3449 #
                0080 ;#                                                              #
                0090 ;################################################################
                0100 ;
                0110 ;
FE1A            0120         ORG  65050       ;STARTADRESSE (Programm nicht frei verschiebbar!"
                0130 ;
                0140 ;
                0150 ;VARIABLENDEFINITION
                0160 ;
5C3D            0170 ERRSP   EQU  23613       ;Systemvar.-Adr. ERR SP, enthält Fehlerrücksprungadr.
                0180 ;
1601            0190 OPEN    EQU  $1601       ;ROM-Routine 'Kanal öffnen'
24FB            0200 STACKIN EQU  $24FB       ;     "      'Auswerten von Ausdrücken'
1C79            0210 STACK2N EQU  $1C79       ;     "      'Berechnen numerischer Ausdrücke'
1E94            0220 UNSTCKA EQU  $1E94       ;     "      'Int.-Zahlen vom Calc.-Stack holen'
                0230 ;
5C92            0240 FORMFLG EQU  23698       ;Hier ist gewählte Druckart binär codiert gespeichert.
                0250 ;
                0260 ;Bedeutung der Bits von FORMFLG (Formatflag):
                0270 ;
                0280 ;Bit 0: Gesetzt, falls X-Streckung
                0290 ;Bit 1: Gesetzt, falls Y-Streckung
                0300 ;Bit 2: Gesetzt, bei Y-Streckung im 2. Druckdurchlauf einer Zeile
                0310 ;Bit 7: Gesetzt, falls Invertierung gewählt
                0320 ;
5C93            0330 ZLNR    EQU  23699       ;Speicherplatz für Zeilennummer
5C94            0340 ZLAN    EQU  23700       ;Speicherplatz für Zeilenanzahl
5C95            0350 SPNR    EQU  23701       ;Speicherplatz für Spaltennummer
5C96            0360 SPAN    EQU  23702       ;Speicherplatz für Spaltenanzahl
5C9E            0370 SBUFF   EQU  23710       ;Beginn des Zwischenspeichers für 8 Bildbytes
                0380 ;
001B            0390 ESC     EQU  27          ;ESC-Code für Druckersteuerung
000D            0400 CR      EQU  13          ;Carriage return (=Zeilenende)
000A            0410 LF      EQU  10          ;Line feed (=Zeilenvorschub)
                0420 ;
                0430 ;---------------------------------------------------------------
                0440 ;
                0450 ;PROGAMMANFANG
                0460 ;
FE1A CD29FF     0470 START   CALL STEST       ;Prüfen, ob dem 'PRINT USR'-Aufruf ',' oder ' ;' folgt!
FE1D 3E00       0480         LD   A,0         ;A-Reg. mit 0 als Wert für Druckformat 'Normale Größe' laden.
FE1F 2009       0490         JR   NZ,FLAG_0   ;Kein Komma oder Semikolon? Dann folgen keine Parameter!
                0500 ;
                0510 ;BRUCKFORMAT AUS PRINT USR-ZEILE HOLEN
                0520 ;
FE21 E7         0530         RST  $20         ;Sonst nächstes Parameter holen aus der BASIC-Zeile
FE22 CDFB24     0540         CALL STACKIN     ;und Auswerten des Ausdrucks. Ergebnis auf Kalk.-Stapel!
FE25 CD941E     0550         CALL UNSTCKA     ;Integerzahl wieder vom Kalkulator-Stapel holen
FE28 E683       0560         AND  $83         ;Nur Bit 7, 1 und 0 einblenden ! ($83 = 10000011)
                0570 ;
FE2A 32925C     0580 FLAG_0  LD   (FORMFLG),A ;Ergebnis (= Druckformat) in FORMFLG laden!
                0590 ;
                0600 ;ZEILENNUMMER UND ZEILENZAHL AUS PRINT USR-ZEILE HOLEN
                0610 ;
FE2D 210018     0620         LD   HL,$1800    ;Standardwerte für Hardcopy-Zeilenzahl $18 (= 24) laden
FE30 22935C     0630         LD   (ZLNR),HL   ;und in Variable 'ZLNR' ablegen!
FE33 2620       0640         LD   H,$20       ;Standardwert für Hardcopy-Spaltenzahl $20 (= 32! laden
FE35 22955C     0650         LD   (SPNR),HL   ;und in Variable 'SPNR' ablegen!
FE38 CD29FF     0660         CALL STEST       ;Weitere Parameter? Dann ist nächstes Zeichen ',' oder ' ;'!
FE3B 2044       0670         JR   NZ,NORM     ;Falls keine vorhanden, bleibt's bei den Standardwerten!
FE3D CD791C     0680         CALL STACK2N     ;Sonst die nächsten zwei Zahlen auf Kalk.-Stapel bringen,
FE40 CD941E     0690         CALL UNSTCKA     ;und die 2. Zahl vom Kalk.-Stapel ins A-Reg. laden!
FE43 32945C     0700         LD   (ZLAN),A    ;Diese als zu druckende Zeilenanzahl in 'ZLAN' bringen.
FE46 A7         0710         AND  A           ;Ist es eine 0? Dann wird Z-Flag gesetzt
FE47 2820       0720         JR   Z,ENDE      ;und der Parametertest ist beendet.
FE49 F5         0730         PUSH AF          ;Sonst Zeilenanzahl merken,
FE4A CD941E     0740         CALL UNSTCKA     ;und 1. Zahl vom Kalk.-Stapel holen und ins A-Reg. laden.
FE4D 32935C     0750         LD   (ZLNR),A    ;Diesen Wert als Nummer der ersten Zeile in 'ZLNR' bringen
FE50 C1         0760         POP  BC          ;Zeilenanzahl vom Stapel ins BC-Reg holen, dann
FE51 80         0770         ADD  B           ;ZLAN und ZLNR addieren = Endzeile (Ergebnis in A-Reg.)!
FE52 382B       0780         JR   C,ERROR     ;Wenn ZLNR + ZLAN > 255, dann Fehlermeldung ausgeben!
FE54 FE19       0790         CP   $19         ;Nur Werte <25 als Endzeile zulassen, darum
FE56 3027       0800         JR   NC,ERROR    ;auch bei Endzeile =>25, Fehlermeldung ausgeben!
FE58 CD29FF     0810         CALL STEST       ;Sonst Test des akt. Zeichens auf "," oder " ;" !
FE5B 2024       0820         JR   NZ,NORM     ;Keins davon? Dann keine Parameter, daher Normal-COPY.
FE5D CD791C     0830         CALL STACK2N     ;Sonst ROM-Routine zur Berechnung num. Ausdrücke aufrufen
FE60 CD941E     0840         CALL UNSTCKA     ;und Int.-Zahl vom Kalk.-Stapel nehmen (Spaltennr.!.
FE63 32965C     0850         LD   (SPAN),A    ;Diese im Speicherplatz für Spaltenanzahl ablegen!
```

```
FE66 A7          0860          AND  A             ;Ist die Spaltenanzahl 0? (d.h. kein Ausdrucken erforderlich!.
FE67 2007        0870          JR   NZ,WEITER     ;Falls nicht, dann weiter!
                 0880 ;
FE69 EB7B3D5C    0890 ENDE     LD   SP,(ERRSP)    ;Sonst Fehlerrücksprungadresse vom Stapel holen
FE6D C3761B      0900          JP   $1B76         ;und über die ROM-Routine )BREAK-Abfrage( zurück ins BASIC!
                 0910 ;
                 0920 ;-------------------------------------------------------------------
                 0930 ;
                 0940 ;SPALTENNUMMER UND SPALTENANZAHL AUS PRINT USR-ZEILE HOLEN
                 0950 ;
FE70 F5          0960 WEITER   PUSH AF            ;Spaltenanzahl auf Stapel retten und
FE71 CD941E      0970          CALL UNSTCKA       ;nächste Int.-Zahl vom Kalk.-Stapel holen (Spaltennummer)
FE74 32955C      0980          LD   (SPNR),A      ;Diese in Speicherplatz für Spaltennummer ablegen.
FE77 C1          0990          POP  BC            ;Nun Spaltenanzahl vom Stapel zurückholen ins BC-Register und
FE78 80          1000          ADD  B             ;dazu Wert aus A-Reg. (=Spaltennummer) addieren.
FE79 3804        1010          JR   C,ERROR       ;Falls Endspaltenwert )255, Fehlermeldung ausgeben!
FE7B FE21        1020          CP   $21           ;Für Endspalte nur Werte (33 zulassen!
FE7D 3802        1030          JR   C,NORM        ;Solange letzte Spaltennr. ( 33 bleibt, alles i.O.!
                 1040 ;
FE7F CF          1050 ERROR    RST  $08           ;Sonst Fehlermeldungsroutine im ROM anfrufen und
FE80             1060          DB   $0A           ;die Meldung "Integer out of range" ausgeben!
                 1070 ;
                 1080 ;-------------------------------------------------------------------
                 1090 ;
FE81 3E03        1100 NORM     LD   A,$03         ;Kanalnummer für Druckerausgabe laden
FE83 CD0116      1110          CALL OPEN          ;und diesen Kanal öffnen!
FE86 3E1B        1120          LD   A,ESC         ;Zur Einstellung des Zeilenabstandes zuerst Code 27 laden
FE88 D7          1130          RST  $10           ;und zum Drucker schicken.
FE89 3E41        1140          LD   A,"A"         ;Dann CHR$ "A" (= Code 65) laden
FE8B D7          1150          RST  $10           ;und ebenso zum Drucker senden!
FE8C 3E08        1160          LD   A,$08         ;Genauso mit Code 8 verfahren. Mit der Codefolge 27, 65, 8
FE8E D7          1170          RST  $10           ;wurde nun der Zeilenabstand auf 8/72 Inch eingestellt!
                 1180 ;
FE8F 3E1B        1190 COPLN    LD   A,ESC         ;Nun in gleicher Weise den Drucker per ESC-Sequenzen
FE91 D7          1200          RST  $10           ;in den Grafikmodus umschalten (siehe Druckerhandbuch)!
FE92 3E4B        1210          LD   A,"K"         ;Dazu ist die Codefolge 27, 75, n nötig! Der Wert n wird im
FE94 D7          1220          RST  $10           ;Folgenden errechnet (=Spaltenzahl in Grafikpunkte umrechnen).
FE95 3A965C      1230          LD   A,(SPAN)      ;Dazu zuerst die Spaltenanzahl ins A-Reg. laden,
FE98 07          1240          RLCA               ;und den Wert 2 mal kreisförmig links rotieren! Aus 32 (00100000)
FE99 07          1250          RLCA               ;wird im 1. Durchgang 64 (01000000), im 2. 128 (10000000).
FE9A 2600        1260          LD   H,0           ;Nun H-Reg. löschen und anschließend das 7. Bit des Wertes in A
FE9C 17          1270          RLA                ;links durchs C-Flag rotieren (1 in C-Flag, A-Reg. wird 00000000)!
FE9D CB14        1280          RL   H             ;H-Reg. links durch C-Flag rotieren (0 in C-Flag, H =00000001)!
                 1290 ;
                 1300 ;Auf diese Weise wurde die Spaltenanzahl mit 8 multipliziert und so
                 1310 ;die Anzahl der zu druckenden Grafikpunkte errechnet.
                 1320 ;
FE9F FDCB5846    1330          BIT  0,(IY+88)     ;Ist Bit 0 von FORMFLG gesetzt? Dann X-Streckung.
FEA3 2803        1340          JR   Z,NOSTR       ;Wenn nicht,wurde Ausgabe im Normalformat gewählt!
FEA5 17          1350          RLA                ;Sonst nochmal A-Reg. links durch C-Flag rotieren und ebenso mit
FEA6 CB14        1360          RL   H             ;H-Reg. verfahren. So wurde die Grafikpunkteanzahl verdoppelt.
                 1370 ;
FEA8 D7          1380 NOSTR    RST  $10           ;Errechnete Punktanzahl (A-Reg.) zum Drucker schicken,
FEA9 7C          1390          LD   A,H           ;den Wert aus dem H-Reg. ins A-Reg. umladen und dann als
FEAA D7          1400          RST  $10           ;Highbyte auch diesen Wert dem Drucker übergeben!
FEAB FD66594     1410          LD   H,(IY+89)     ;H-Reg. mit Wert für ZLNR (Zeilennummer) laden.
FEAE FD6E5B      1420          LD   L,(IY+91)     ;L-Reg. mit Wert für SPNR (Spaltennummer) laden.
FEB1 FD465C      1430          LD   B,(IY+92)     ;B-Reg. mit Wert für SPAN (Spaltenanzahl) laden.
                 1440 ;
FEB4 E5          1450 ZEILE    PUSH HL            ;ZLNR- und SPNR-Werte auf den Stapel retten und
FEB5 C5          1460          PUSH BC            ;den SPAN-Wert ebenso.
FEB6 CD30FF      1470          CALL SCADR         ;Nun Adr. des obersten Bytes der PRINT-Position berechnen.
FEB9 119E5C      1480          LD   DE,SBUFF      ;DE-Reg. mit Adr. des Zwischenspeichers für 8 Bildbytes
FEBC 0608        1490          LD   B,8           ;und B-Reg. als Schleifenzähler mit 8 laden.
FEBE FDCB5856    1500          BIT  2,(IY+88)     ;Bit 2 von FORMFLG gesetzt? Dann Y-Streck. 2. Durchlauf!
FEC2 2804        1510          JR   Z,READ        ;Wenn nicht, dann wird noch der 1. Durchlauf abgearbeitet.
FEC4 24          1520          INC  H             ;2. Durchlauf in Arbeit? Dann das High-Byte der Pixelzeilenadr.
FEC5 24          1530          INC  H             ;um 4 erhöhen, da wegen der Y-Streckung im 1. Durchlauf
FEC6 24          1540          INC  H             ;nur die ersten 4 Bildbytes jeweils doppelt ausgegeben wurden
FEC7 24          1550          INC  H             ;und nun die restlichen 4 bearbeitet werden müssen!
                 1560 ;
FEC8 7E          1570 READ     LD   A,(HL)        ;Bildbyte aus ermittelter Zeilenadr. ins A-Reg. holen
FEC9 12          1580          LD   (DE),A        ;und dieses in SBUFF ablegen.
FECA 1C          1590          INC  E             ;SBUFF-Adresse auf nächste setzten und
FECB FDCB584E    1600          BIT  1,(IY+88)     ;prüfen, ob Y-Streckung erforderlich ist.
FECF 2802        1610          JR   Z,NOSTY       ;Nein? Dann Doppelausgabe des letzten Bildbytes übergehen!
FED1 12          1620          LD   (DE),A        ;Sonst letztes Bildbyte nochmal in SBUFF ablegen
FED2 1C          1630          INC  E             ;und SBUFF-Adresse erneut erhöhen!
                 1640 ;
FED3 24          1650 NOSTY    INC  H             ;Bildbyte-Adresse auf nächste setzen
FED4 10F2        1660          DJNZ READ          ;und weitere Bildbytes umladen.
FED6 1608        1670          LD   D,$08         ;D-Reg. für äußere Schleife mit 8 laden (Pixelspaltenzähler).
FED8 219E5C      1680 OUT1     LD   HL,SBUFF      ;Dann Adresse der Zwischenablage SBUFF in HL bringen.
FEDB 0608        1690          LD   B,$08         ;B-Reg. für innere Schleife mit 8 laden (Pixelzeilenzähler!
                 1700 ;
```

```
FEDD 5E          1710 OUT2   LD   E,(HL)       ;und Zeichen aus SBUFF ins E-Reg. holen.
FEDE CB13        1720        RL   E            ;Nun den Operanden (Wert in E) links durchs C-Flag rotieren, dann
FEE0 17          1730        RLA               ;A-Reg. links durchs C-Flag rotieren, um Flag-Wert zu speichern!
FEE1 73          1740        LD   (HL),E       ;Veränderten Code aus E-Reg. in SBUFF zurückschreiben
FEE2 2C          1750        INC  L            ;und SBUFF-Adr. erhöhen, d.h. auf nächstes Zeichen setzen.
FEE3 10F8        1760        DJNZ OUT2         ;Weiter in der inneren Schleife, bis 8. Durchlauf beendet!
FEE5 FDCB587E    1770        BIT  7,(IY+88)    ;Ist Bit 7 von FORMFLG gesetzt (d.h. = 1)?
FEE9 2801        1780        JR   Z,NOINV      ;Falls nicht, ist Z-Flag =0, also keine Invertierung!
FEEB 2F          1790        CPL               ;Sonst alle Bits im A-Reg. invertieren (= Komplement bilden).
                 1800 ;
FEEC F5          1810 NOINV  PUSH AF           ;Nun den ermittelten Pixelspaltenwert auf den Stapel legen
FEED D7          1820        RST  $10          ;und anschließend zum Drucker schicken!
FEEE F1          1830        POP  AF           ;Pixelspaltenwert vom Stapel zurückholen und
FEEF FDCB5846    1840        BIT  0,(IY+88)    ;Bit 0 von FORMFLG prüfen. Nicht gesetzt (=0)? Dann ist
FEF3 2801        1850        JR   Z,NOSTX      ;Z-Flag gesetzt, also keine X-Streckung gewünscht!
FEF5 D7          1860        RST  $10          ;Sonst den Pixelspaltenwert nochmal zum Drucker schicken!
FEF6 15          1870 NOSTX  DEC  D            ;Nun den Pixelspaltenzähler (äußere Schleife) erhöhen
FEF7 20DF        1880        JR   NZ,OUT1      ;und zurück, bis D = 0 ist!
FEF9 C1          1890        POP  BC           ;Dann Spaltenanzahl (SPAN) und
FEFA E1          1900        POP  HL           ;Zeilen- und Spaltennummer (ZLNR / SPNR) vom Stapel holen,
FEFB 2C          1910        INC  L            ;SPNR auf neue Spalte setzen und
FEFC 10B6        1920        DJNZ ZEILE        ;die nächsten 8 Bildbytes holen, bearbeiten und drucken!
FEFE 3E0D        1930        LD   A,CR         ;Sind alle Spalten (SPAN = 0) ausgegeben, Carriage return
FF00 D7          1940        RST  $10          ;zum Drucker schicken und
FF01 3E0A        1950        LD   A,LF         ;den Line feed Code ebenso. Damit wird die Ausgabe einer
FF03 D7          1960        RST  $10          ;Druckzeile ausgelöst.
FF04 21925C      1970        LD   HL,FORMFLG   ;Adresse von FORMFLG ins HL-Reg. laden und
FF07 CB4E        1980        BIT  1,(HL)       ;prüfen, ob Bit 1 davon gesetzt ist.
FF09 2809        1990        JR   Z,NOYST      ;Nein? Dann keine Y-Streckung! Also weiter bei NOYST.
FF0B CB56        2000        BIT  2,(HL)       ;Doch? Dann Bit 2 prüfen (auf 2. Durchlauf bei Y-Streckung)
FF0D CBD6        2010        SET  2,(HL)       ;und dieses nun setzten (= 1)!
FF0F CA8FFE      2020        JP   Z,COPLN      ;War es noch ungesetzt? Dann ist erst 1 Durchlauf beendet.
FF12 CB96        2030        RES  2,(HL)       ;Sonst war es schon der 2., dann Bit 2 von FORMFLG löschen!
                 2040 ;
FF14 FD3459      2050 NOYST  INC  (IY+89)      ;Nun Zeilennummer (ZLNR) auf nächsten Wert erhöhen
FF17 FD355A      2060        DEC  (IY+90)      ;und Zeilenanzahl (ZLAN) entsprechend verringern!
FF1A C28FFE      2070        JP   NZ,COPLN     ;Nun neue Zeile bearbeiten, solange ZLAN <> 0!
FF1D 3E1B        2080        LD   A,ESC        ;Ist die letzte Zeile gedruckt worden (ZLAN = 0),
FF1F D7          2090        RST  $10          ;dann mit der Steuersequenz 'ESC "A" 12' den Zeilenabstand
FF20 3E41        2100        LD   A,"A"        ;wieder auf den ursprünglichen Wert von 12/72 Zoll
FF22 D7          2110        RST  $10          ;d.h. 1/6 Zoll zurücksetzten!
FF23 3E0C        2120        LD   A,12         ;Damit ist der Drucker wieder für den Ausdruck von Text
FF25 D7          2130        RST  $10          ;mit normalem Zeilenabstand eingestellt.
FF26 C369FE      2140        JP   ENDE         ;Nun kann die Rückkehr ins BASIC-Programm eingeleitet werden.
                 2150 ;
                 2160 ;-------------------------------------------------------------------
                 2170 ;
                 2180 ;PRINT USR-ZEILE AUF ' ;' ODER ',' PRÜFEN (= TRENNZEICHEN BEI PARAMETER
                 2190 ;
FF29 DF          2200 STEST  RST  $18          ;Akt. Zeichen aus Programm oder Eingabe holen
FF2A FE2C        2210        CP   $2C          ;und vergleichen, ob es ein Semikolon ist.
FF2C C8          2220        RET  Z            ;Dann zurück ins Programm!
FF2D FE3B        2230        CP   $3B          ;Oder ist es vielleicht ein Komma?
FF2F C9          2240        RET               ;Nun auf jeden Fall zurück!
                 2250 ;
                 2260 ;-------------------------------------------------------------------
                 2270 ;
                 2280 ;PIXELZEILENANFANGSADR. EINER NEUEN ZEILEN-/SPALTENPOSITION ERRECHNEN
                 2290 ;
FF30 F5          2300 SCADR  PUSH AF           ;Augenblicklichen A-Reg.-Inhalt auf Stapel retten
FF31 7C          2310        LD   A,H          ;und A-Reg. mit Zeilennummer (ZLNR) laden und
FF32 E607        2320        AND  $07          ;Bit 0-2 einblenden, also nur Werte von 0-7 zulassen.
FF34 0F          2330        RRCA              ;Nun die 3 Bits rechts kreisförmig rotieren, d.h. alle der Reihe
FF35 0F          2340        RRCA              ;nach ins C-Flag bringen und ins Bit 5-7 rotieren. So steht im
FF36 0F          2350        RRCA              ;A-Reg. nun die Pixeladr. (low byte) der Anfangspixelzeile!
FF37 B5          2360        OR   L            ;Spaltennummer (SPNR) nun in diese Pixelzeilenadr. einblenden.
FF38 6F          2370        LD   L,A          ;Das Ergebnis als Pixelzeilenanfang ins L-Reg. ablegen und
FF39 7C          2380        LD   A,H          ;die Zeilennummer (ZLNR) ins A-Reg. zurückholen.
FF3A E618        2390        AND  $18          ;Bit 3-4 einblenden, um event. Zeile 8-23 zu berücksichtigen,
FF3C C640        2400        ADD  $40          ;$40 add. zum Ergebnis (=Highbyte des Bildspeicheranfangs!
FF3E 67          2410        LD   H,A          ;und den neuen Wert als Pixelzeilenanfangsadr. ins H-Reg. laden
FF3F F1          2420        POP  AF           ;Ursprünglichen Inhalt des A-Reg. vom Stapel zurückholen
FF40 C9          2430        RET               ;und zurück!

0       61A8  ERRSP   5C3D  OPEN    1601  STACKIN 24FB  STACK2N 1C79  UNSTCKA 1E94  FORMFLG 5C92  ZLNR    5C93  ZLAN    5C94
SPNR    5C95  SPAN    5C96  SBUFF   5C9E  ESC     001B  CR      000D  LF      000A  START   FE1A  FLAG_0  FE2A  ENDE    FE69
WEITER  FE70  ERROR   FE7F  NORM    FE81  COPLN   FE8F  NOSTR   FEA8  ZEILE   FEB4  READ    FEC8  NOSTY   FEB3  OUT1    FED8
OUT2    FEDD  NOINV   FEEC  NOSTX   FEF6  NOYST   FF14  STEST   FF29  SCADR   FF30
```

Neues für DTP: Datenpacket II

Decorpacks: Die Nummern 70 bis 96 sind dazugekommen. Rund 1400 Bilder sind es damit zusammen mit Datenpacket I.
Cornerpacks: von 224 bis 327. Das Thema: der gesetzte Rahmen.
Titelschriften: insgesamt 87, davon 46 mit lettermat, 41 "echte" Fnt6-Fonts mit deutschen Sonderzeichen, wo immer sinnvoll.
ca! löscht alle Files aus der Liste,außer FXN,Abfrage Y/N,Break, Anzeige des Files.
del! räumt alle Graphikfiles ab,deren Name mit Asterix (*) beginnt, Break möglich.
ex! entfernt alle Files vom Typ "Bild" aus der Liste.Y/N,BREAK, Anzeige.
edibreit! Der Fnt3 Editor, auf 32 Dot verbreitert.
edimaus! Der normale Editor für AMX-Maus.
edimat! Konvertieren von 8x8 Zeichensätzen in Fnt3 für AMX-Maus eingerichtet.
screen! Expandieren von DTP-Graphik in SCREEN$. Hardcopies von Liste, Katalog (MD und DISK),Ausschnitte aus Texten mit Befehlszeilen, Steuerzeichen usw.
tyg! ein auf 23 Graphikplätze erweiterter typeliner! mit Sparttaste für Farbband und Druckkopf (Esc "s" 1).
dokufont! Dokumentieren von Fnt3-Sätzen,Anfertigen von Entwurfsgittern.
roll! Pixelweises Scrollen von DTP-Graphik in alle vier Richtungen.
col! Aufrastern von DTP-Graphik, Merge-Funktion für Groundpacks, Unterlegen von Schriftbildern, für Texturen, Hintergrund-Bearbeitung.
*** digi! liest mit dem Videoface von Robotronik Videosignale von Fernseher, Videorekorder, Camcorder, anderen Computern bei freier Ausschnittwahl direkt in DTP-Graphikfile s ein. Läuft mit dem DTP.
lettermat! Schriftprogramm, besonders für alte Schriften mit Ligaturen. Beliebige Zahl von Sonderzeichen, keine Einschränkung durch den Tastaturtreiber, Aufruf der Zeichen über ihren Namen.
font! für Fnt6-Sätze, mit deutschen Sonderzeichen, Spatien, Meßgitter.
*** headliner! intern mit headlin.d bezeichnet; deutsche Sonderzeichen.
*** mons! Der Dissassembler von Hisoft im DTP. Für GENS ist ja der Wordmaster von Haus aus eingerichtet.
* serbas! Konvertiert serielle Dateien in tabulierte Textfiles.
* bas! Konvertiert mit "clearcode" Basiclistings zur Verwendung mit dem typeliner.
** clearcode konvertiert Basic-Listings in serielle Dateien.
** Selbstständiges mc-Programm" Fontmaus"; Fnt6-Editor für AMX Maus und Handbetrieb. Deutsche Sonderzeichen!
** Fontkatz: Fnt6-Editor wie "Fontmaus",aber mit der Möglichkeit,normale (8*8) Zeichensätze einzuladen. Diese können hier in Fnt-6 Zeichensätze umgewandelt werden.
Kalendermacher: 8 selbständige Basic & mc-Programme generieren DTP-Textfiles zur individuellen Gestaltung von Kalendern mit dem typeliner! Einbinden von Kalenderteilen in die Textverarbeitung in Anlehnung an den Text-o- Mat. Anwenderpaket "kalpge" mit Kalenderschriften und Demo-pages.
*** WM-Code für Opus mit deutschen Menü's.

Legende:
* für MD und Diskettensysteme, die serielle Dateien vom Specci-Printfile-Typver arbeiten können (z.B. TW3, Masterfile, THE WRITER u sw). IF 1. ROM 1.1 und 1.2-Version.
** bedingtes Copyright (z.B. Routinen aus alten Zeitschriften).
*** Copyright. Abgabe nur an User, die ihre Lizenz z.B. durch Rechnungskopie nachgewiesen haben.

Angebotspreis: Datenpacket II auf 2 OPUS-720 oder 7 OPUS-187 Disketten oder 3 Kassetten. DM 50,-, für Clubmitglieder 30,-
Weiter ist noch lieferbar: Datenpacket I auf 8 OPUS-72 oder 25 OPUS-172 Disketten oder 6 Kassetten.DM 100, für Mitglieder 50,-
Kassetten werden mit 1,-, Disketten mit 1,20 berechnet, Porto n. Anfall.

H.Hartig
8938 Buchloe

Zeichensätze für Texteditoren:

von H.Hartig

Die meisten Zeichensätze für den Editor von Textprogrammen des Spectrum sind schlecht lesbar und haben oft noch nicht einmal die deutschen Sonderzeichen oder diese sind an falscher Stelle oder fast unkenntlich.
Der folgende Artikel soll nun zeigen, wie einfach jeder selbst eigene Zeichensätze entwickeln bzw. die bestehenden ändern kann.
Wenn 64 Zeichen in eine Zeile gehen sollen, dann stehen uns für ein Zeichen nur 3 Pixelspalten zur Verfügung, sonst kleben die Buchstaben aneinander. Ausnahmsweise kann man die Großbuchstaben oder einzelne Sonderzeichen wie &, ', § mit 4 Pixelspalten darstellen, dann kann man aber Worte aus Großbuchstaben schlecht lesbar.
Mit 3 Pixelspalten und einer Höhe von 7 bzw. 6 Pixel ist aber die Möglichkeit, einen Buchstaben lesbar darzustellen, schon recht begrenzt und der Versuch, gerundete Zeichen darzustellen, in dem man an den Ecken Pixel ausläßt, führt zu schlechter Lesbarkeit. Setzen wir nun an die Ecken auch Pixel, so wird die Zeichendarstellung meistens viel besser. Wegen der Kleinheit der Buchstaben rundet unser Auge bzw. unser Gesichtssinn die Ecken ab, auch die meisten Drucker tun dies, sodaß das Schriftbild gefälliger wird.
Das Rezept ist also einfach und wenn wir nun auch noch wissen,wo im Maschinencode die Zeichensätze sitzen, können wir diese verändern. Viele haben ja schon einen Zeichensatzgenerator, mit dem das leicht zu machen ist. Hier ist die Lage der Zeichensätze:

		Speicherplatz f.d.Anfang des Zeichensatzes:	
Bei	Tasword II	61184	das Zeichen sitzt rechts im Raster
	Tasword III	38400	(im Codeteil TASTABLE) " "
	Tasword 128	54272	" "
	Tasword 128+2	54528	" "
	Wordmaster	64512	(Zeichen ist) doppelt "
	THE WRITER 48	56520	links "
	THE WRITER 128	35708	" "

.jeweils 768 bytes.

Für andere Programme kann ich auf Anfrage bzw. Einsendung des M.C. Teils Auskunft geben oder eine Änderung vornehmen.
Die Zeichensätze für obige Programme gebe ich zusammen mit einem, auch zum Auffinden besonders geigneten Zeichengenerator an Herrn Halbritter für die Programmbibliothek.
Der Generator beinhaltet zudem 65 normale (8*8 Pixel) Zeichensätze (jeweils 35 im Speicher und sofort bearbeitbar),sowie Download-Möglichkeit für geeignete Drucker (Epson/kompat.)
Hier noch ein paar Mustersätze f. Editor:

```
150
ein Musterdruck mit Editorzeichensatz:ABCDEFGH abcdefgh ÄÖÜäöüß
153
ein Musterdruck mit Editorzeichensatz:ABCDEFGH abcdefgh ÄÖÜäöüß
156
ein Musterdruck mit Editorzeichensatz:ABCDEFGH abcdefgh ÄÖÜäöüß
159
ein Musterdruck mit Editorzeichensatz:ABCDEFGH abcdefgh ÄÖÜäöüß
162
ein Musterdruck mit Editorzeichensatz:ABCDEFGH abcdefgh ÄÖÜäöüß
165
ein Musterdruck mit Editorzeichensatz:ABCDEFGH abcdefgh ÄÖÜäöüß
168
ein Musterdruck mit Editorzeichensatz:ABCDEFGH abcdefgh ÄÖÜäöüß
171
ein Musterdruck mit Editorzeichensatz:ABCDEFGH abcdefgh ÄÖÜäöüß
174
ein Musterdruck mit Editorzeichensatz:ABCDEFGH abcdefgh ÄÖÜäöüß
177
ein Musterdruck mit Editorzeichensatz:ABCDEFGH abcdefgh ÄÖÜäöüß
118800
eeiinn MMuussttrrddrruucckk mmiitt EEddiittoorrzzeeiicchhrennssaattzz::AABBCCDDEEFFGGHH aabbccddeeffgghh ÄÄÖÖüüßßööüüßß
183
ein Musterdruck mit Editorzeichensatz:ABCDEFGH abcdefgh ÄÖÜäöüß
```

Hubert Kracher
Kirchdorfer Str. 5
8201 Großholzhausen

****
* DOUBLE 'H' SOFTWARE *
****

Harald Lack
Heidenauer Str. 5
8201 Raubling

Hallo Spectrum-Freunde!!

Wie wohl jeder ersthafte Anwender weiß, gibt es eine Menge von Utilities, die die Anzahl der möglichen Zeichen pro Zeile von 32 auf 64 erhöhen und damit die darstellbare Informationsmenge verdoppeln. Für all diejenigen, die eine einfach zu bedienende Routine suchen gibt es heute ein relativ kurzes Maschinenprogramm als vollständiges Assemblerlisting. Das Programm stammt von Volker Ziegler und hat mitlerweile bereits einige Jahre auf dem Buckel. Nichts desto trotz ist es auch heute noch sehr effizient einsetzbar. Einigen von Euch wird das Programm vielleicht bekannt vorkommen, den unser Mit-User Karl-Heinz Meander hat sich bereits im Info 3/91 einmal kurz mit dem Programm beschäftigt. Er veröffentlichte damals den Hexdump dieser Routine, die bei einigen Usern recht gut ankam. Wir möchten nun das gesamte Programm als Assemblerlisting vorstellen und nochmals kurz die wichtigsten Punkte beschreiben, damit ihr Euch das Blättern in diversen Infos erspart.
Wer das Assemblerlisting näher betrachtet wird feststellen, daß das Programm eine relativ große Anzahl von Leerzeilen oder Kommentarzeilen enthält, was das Lesen des Listings auch für den etwas ungeübteren User einfach gestal tet. Natürlich braucht man diese Zeilen bei der Verarbeitung mittels eines Assemblers nicht einzutippen, da sie sowieso nicht verarbeitet werden. Jetzt aber ein paar Worte zum Programm selbst. Geladen wird das assemblierte Programm, das ja nun als ablauffähiger Maschinencode zur Verfügung steht, wie jedes andere MC-Programm auch mit CLEAR 63999: LOAD "" CODE. Danach muß unbedingt folgende Zeile eingegeben werden: 1 REM xxxxx. Da die Schrift mit 64 Zeichen mittels des Befehles PRINT #4; aufgerufen wird benötigt man diese 5 Byte (xxxxx) als Channel Data (Kanal-Daten). Die Routine wird einfach mit dem Befehl RANDOMIZE USR 64000 gestartet. Wenn man sich jetzt die Zeile 1 ansieht, erkennt man, daß sie sich verändert hat. Sie darf jetzt unter keinen Umständen mehr verändert werden und sollte deshalb mittel POKE 23756,0 vor dem Editieren geschützt werden. Um die 64 Zeichen pro Zeile zu nutzen, kann man jetzt alle BASIC PRINT Befehle auch mit PRINT #4 aufrufen und damit die 64 Zeichen anwenden. Das hat unter anderem den Vorteil, daß diese Routine ganz einfach aus BASIC heraus aufgerufen werden kann und keine Unterprogramme angesprungen werden müßen. Außerdem kann man weiterhin die Funktionen INK, PAPER, BRIGHT, FLASH, INVERSE und OVER verwenden. Bei AT und TAB ist zu beachten, daß jetzt Spaltenzahlen zwischen 0 und 63 möglich sind. Das eigentliche Maschinenprogramm benötigt einschließlich Zeichensatz nur 729 Byte und die kann man wohl bei der Programmiererei sicherlich verschmerzen. Da aber bei 64 Zeichen je Zeile nur jeweils 8 halbe Byte pro Zeichen benutzt werden, liegt es nahe, den Zeichensatz zu komprimieren. Daraus folgt, daß man pro 8 Byte 2 Zeichen unterbringen kann. Diese Maßnahme spart 384 Byte, da der normale Zeichensatz ja auch bekannterweise 768 Byte umfaßt. Und man erhält die Möglichkeit, die Blockgrafikzeichen selbst zu definieren. Bei dieser Programmversion hat es der Autor so vorgesehen, daß auf den Grundtasten 1 - 8 die normalen und auf CAPS 1 - 8 die selbstdefinierten Grafikzeichen stehen. Dieses Programm bietet auch die Möglichkeit die USR-Grafikzeichen auszudrucken. Bitte beachten, daß im Aseemblerlisting nur ein Platzhalter für den Zeichensatz vorhanden ist. Die Werte müßen der Dezimaldump Tabelle entnommen werden und in das Unterprogramm gepoket werden. Die Dez-Dump-Liste gibt es dann am Ende des Assemblerlistings in einem der nächsten Infos

inklusive einer praktischen Laderoutine in BASIC.

Bis dahin viel Spaß beim Experimentieren mit dem Spectrum.

```
                0010 ;          "Z 80 Assembler Program"
                0020 ;          "Assembled by EDITAS"
                0022 ;
                0023 ;
                0024 ;          "**********************"
                0025 ;          "*                     *"
                0030 ;          "* 64 Zeichen / Zeile *"
                0040 ;          "*                     *"
                0050 ;          "**********************"
                0060 ;
                0070 ;
                0080 ;
                0090 ;          "Version 2.11 by V.Ziegler"
                0100 ;          "*************************"
                0110 ;
                0120 ;
FA00            0130        ORG  64000
                0140 ;
                0150 ;
5C7B            0160 UDG    EQU  23675
5CD0            0170 CHDAT  EQU  23760
5C91            0180 PFLAG  EQU  23697
5C51            0190 CHURC  EQU  23633
5BF4            0200 ZCHEN  EQU  23540
                0210 ;
                0220 ;          "= Ende Printerbuffer"
                0230 ;
5C0E            0240 TVDTL  EQU  23566
5C0F            0250 TVDTH  EQU  23567
0A80            0260 POCHG  EQU  00A80H
0A9B            0270 POAT1  EQU  00A9BH
                0280 ;
0010            0290 INK    EQU  16
0016            0300 AT     EQU  22
0017            0310 TAB    EQU  23
                0320 ;
                0330 ;
                0340 ;
                0350 ;          "INIT - Routine"
                0360 ;          "==============="
                0370 ;
FA00 3E1B       0380 INIT   LD   A,27
FA02 321E5C     0390        LD   (23582),A
                0400 ;
FA05 2111FA     0410        LD   HL,ITXT
FA08 11D05C     0420        LD   DE,CHDAT
FA0B 010500     0430        LD   BC,5
FA0E EDB0       0440        LDIR
FA10 C9         0450        RET
                0460 ;
                0470 ;
FA11 17FA       0480 ITXT   DEFW START
FA13 08         0490        DEFB 08
FA14 00         0500        DEFB 00
FA15 53         0510        DEFB "S"
                0520 ;
                0530 ;
FA16 00         0540 PSALT  DEFB 0
                0550 ;
                0560 ;
                0570 ;
                0580 ;          "Hauptprogramm"
                0590 ;          "=============="
                0600 ;
FA17 CD030B     0610 START  CALL 00B03H
                0620 ;
```

```
                0650 ;
FA1A FE0D       0660        CP   00DH
FA1C CCD5FA     0670        CALL Z,PCR
                0680 ;
FA1F FE10       0690        CP   INK
FA21 DAF409     0700        JP   C,009F4H
                0710 ;
FA24 FE16       0720        CP   AT
FA26 DADBFA     0730        JP   C,INKOV
                0740 ;
FA29 FE18       0750        CP   TAB+1
FA2B DAE4FA     0760        JP   C,ATTAB
                0770 ;
FA2E FEA5       0780        CP   165
FA30 D2F409     0790        JP   NC,009F4H
                0800 ;
                0810 ;
                0820 ;          "normales Zeichen"
                0830 ;          "================"
                0840 ;
FA33 C5         0850        PUSH BC
FA34 0199FA     0860 CHAR2  LD   BC,ZCH64-128
FA37 FE90       0870        CP   144
FA39 3030       0880        JR   NC,PH0
                0890 ;
                0900 ;          "Zeichen aus komprimierten"
                0910 ;          "Zeichensatz"
                0920 ;          "=========================="
                0930 ;
FA3B E5         0940        PUSH HL
FA3C 1E00       0950        LD   E,0
FA3E 1F         0960        RRA
FA3F CB13       0970        RL   E
                0980 ;
                0990 ;          "Carry-Bit in E"
                1000 ;
FA41 87         1010        ADD  A
FA42 6F         1020        LD   L,A
FA43 2600       1030        LD   H,0
                1040 ;
                1050 ;          "E=0=> links"
                1060 ;          "E=1=> rechts"
                1070 ;          "============"
                1080 ;
FA45 29         1090        ADD  HL,HL
FA46 29         1100        ADD  HL,HL
FA47 09         1110        ADD  HL,BC
                1120 ;
                1130 ;          "HL = Characterposition"
                1140 ;
FA48 AF         1150        XOR  A
FA49 BB         1160        CP   E
FA4A 11F45B     1170        LD   DE,ZCHEN
FA4D D5         1180        PUSH DE
FA4E 0608       1190        LD   B,8
FA50 280C       1200        JR   Z,PHL
                1210 ;
                1220 ;          "Zeichen von rechts nach"
                1230 ;          "links verschieben"
                1240 ;          "======================"
                1250 ;
FA52 7E         1260 L1     LD   A,(HL)
FA53 87         1270        ADD  A
FA54 87         1280        ADD  A
FA55 87         1290        ADD  A
FA56 87         1300        ADD  A
```

Hallo Spectrum & Clubfreunde!

Durch den Ausfall des Januar-Hefts (Hallo Rolf! wie gehts!) bin ich in meinen Beiträgen ein wenig durcheinander gekommen. Hab also wie man so schön sagt, den Faden verloren.

Daher in aller Schnelle, ein Kalender-Programm; weil die momentan wieder sehr gefragt sind. Man kann beobachten, daß sogar beim SAM nicht davor halt gemacht wird. Und deshalb habe ich mich entschloßen, eines dieser Programme als Beitrag zum Clubinfo vorzustellen.

Erstellt wurde es mit Beta Basic in CSIZE 8,16; um so den ganzen Bildschirm als Kalenderblatt benutzen zu können.

Die Hardcopy entspricht der PLUS D Syntax und dürfte eigentlich sehr leicht an andere Hardcopy-Befehle angepasst werden können.

Tja, und das wars auch schon für diesmal.

Und bevor ich's vergesse:

SUCHE:

Das Programm V-NOTCH!!

Karl-Heinz Meander
Am Schloßberg 19
8431 Hohenfels

```
  4 POKE §6,1
    LPRINT CHR$ 27;CHR$ 64;
  5 POKE §6,Ø
 10 INPUT "Year 19";y
 20 LET d=1
 30 FOR a=1901 TO 1900+y-1
 40  IF a/4=INT (a/4) AND a/1000<>INT (a/1000) THEN LET d=d+366
      GO TO 60
 50  LET d=d+365
 60 NEXT a
 70 REM "D" IST ANZAHL DER TAGE SEIT MONTAG DEN 1.JAN.1900
 80 LET e=d-7*INT (d/7)
 85 RANDOMIZE USR 58419
    CLS #
 90 CSIZE 8,16
 95 RESTORE
 97 INK 1
100 FOR m=1 TO 12
110  READ m$,l
115  IF y/4=INT (y/4) AND m$="February" THEN LET l=29
120  PRINT TAB 10;m$;" ";1900+y
130  PRINT "_____"
140  PRINT "Mon Tue Wed Thu Fri Sat Sun"
150  PRINT "_____"
160  FOR a=1 TO l
170   PRINT TAB e*4; USING "##";a;
180   LET e=e+1
190   IF e=7 THEN LET e=Ø
```

```
200  NEXT a
205  PRINT
210  PRINT "_____________________________________"
220  GET p
222  IF p=12 THEN SAVE SCREEN$
      REM HARDCOPY
225  CLS
230 NEXT m
235 CSIZE 0
240 STOP
250 DATA "January",31,"February",28,"March",31,"April",30,"May",31,"June",30,"J
    uly",31,"August",31,"September",30,"October",31,"November",30,"December",31
```

Verkaufe folgende Teile aus meiner Spectrum-Sammlung:

SPECTRUM +3, eingebautes Disketten-LW, Druckerport, RS 232, Midi, RGB,unbenutzt u. orginalverpackt	260,-DM
SPECTRUM 128 K, orginal, guter Zustand	190,-DM
BETADISK-Interface 5.03 für Spectrum 48k und 128k	150,-DM
Disketten-Laufwerk, 5 1/4, 2* 80Tr., für Betadisk, Disciple, Opus etc., incl. Geh. u. Netzteil	150,-DM
SPECTRUM -Plus , Vierfach-ISO-ROM (Backup-, Monitor-, Basic-Toolkit- und Normal-ROM), Monitoranschluß	150,-DM
Currah Microspeech, Sprache und Sound aus dem TV-Lautsprecher bei allen Programmen	40,-DM
Interface 1, kaum benutzt	50,-DM
Vierfach-ISO-ROM für IF 1 u. Disciple, mit Backup-, Monitor, Basic-Toolkit- u. Normalrom, absturzfreie Umschaltung	40,-DM
Soundmodul 3-Kanal mit eingebautem Lautsprecher incl. Musikprogramm AMADEUS (Orginal)	90,-DM
TRI-STEP, Steuerinterface für den Spectrum, treibt bis zu 7 Ausgänge (z.B. Lämpchen) oder 3 Schrittmotore (z. B. XY-Fahrtisch) Spannungsversorgung über den Spectrum bis 0,8 A, exterener Spannungsanschluß bis 3 A. mit Beispielsoftware in Basic, Gerät fertig aufgebaut und getestet, mit Software zum Ansteuern eines XY-Tisches. Auch Eisenbahnsteuerung möglich !	50,-DM

Alle Preise VB zzgl. Portokosten

Hartmut Schwinty, Liebigstr. 5, 4600 Dortmund 1, Tel. 0231/123109

Ludwig Halbritter
Augsburger Str. 4
8935 Fischach
Tel.:08236/495

Fischach, 18.02.92

Hallo Spectrumfreunde,

die PD-Bibliothek nimmt weiterhin an Umfang zu. Programmkassetten erhielt ich von Josef Schumacher, Herbert Hartig und Klaus Gürtler. Da weiterhin auch Bestellungen eingehen, wird es noch Wochen dauern bis ich alle eingesandten Programme katalogisiert habe.
Unklar war einigen, ob unter dem Begriff "Zugriff auf die PD-Bibliothek" die Bestellung eines oder mehrerer Programme möglich ist. Ein Zugriff auf die Bibliothek kann mehrere Programme bis zum Inhalt einer Kassette umfassen, es ist also mit "Zugriff auf die PD-Bibliothek" ein postalischer Vorgang gemeint.

Karl-Heinz Meander: Hocherfreut über eine "neue" Snap-Funktion habe ich das kurze Programm abgetippt, doch leider funktioniert es nicht richtig. Beim Drücken des Snapshot Button + der Taste 6 fährt der Druckkopf ganz nach links, dann ist Ende der Druckervorstellung, doch auf dem Bildschirm erscheint an der Cursor-Position die 6. Drücke ich dann nochmals den Snapshot Button und danach aber die Taste 1, so erfolgt der Bildschirmausdruck mittig. Feststellen möchte ich, daß mir keine Abtippfehler unterlaufen sind und mein Drucker EPSON-kompatibel ist. Wo hat sich hier ein Fehler eingeschlichen?

Bis zum nächsten Info

Ludwig

Franz Large
Gartenstraße 2
D-6945 Hirschberg

Hirschberg, 27.1.1992

Hallo Spectrum-User!

Bei der Eingabe von Listings aus dem Userclub Info habe ich mich öfter über die "Druckerzeichen" geärgert, welche in ausgedruckten Programmen statt der "normalen" Spectrumzeichen stehen und die zu Übertragungsproblemen führen können. Ein Abhilfe soll das nachfolgende Programm bringen, welches den Spectrumzeichensatz an den Drucker "übergibt". Danach entspricht das ausgedruckte Listing (fast) der Ausgabe auf dem Schirm. Das Zeichen 127 allerdings ist auf diese Weise nicht zu ändern, da es der Drucker (star NL-10) als "DELETE" (löschen des letzten zum Drucker übertragenen Zeichens) interpretiert.

Franz

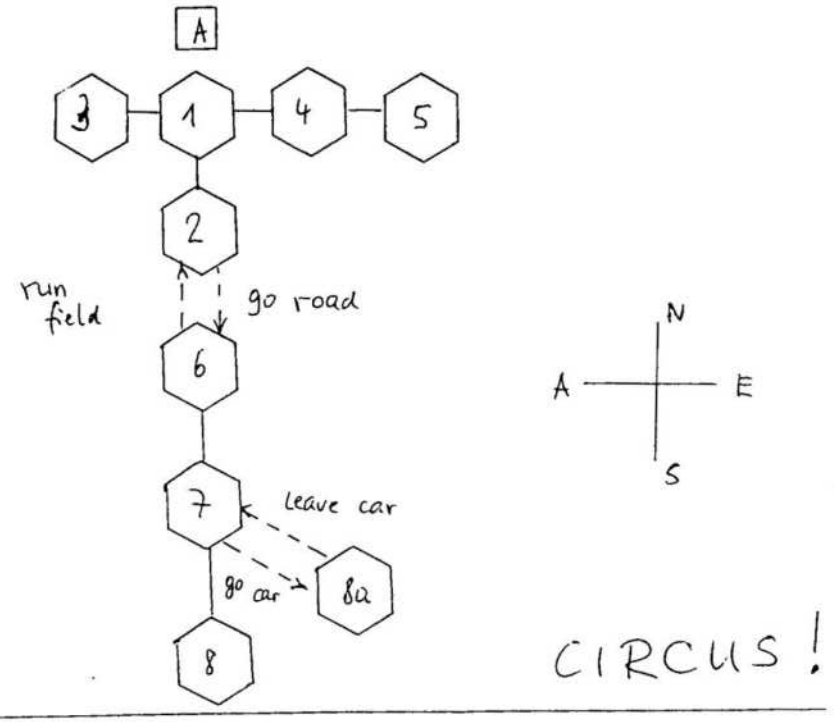
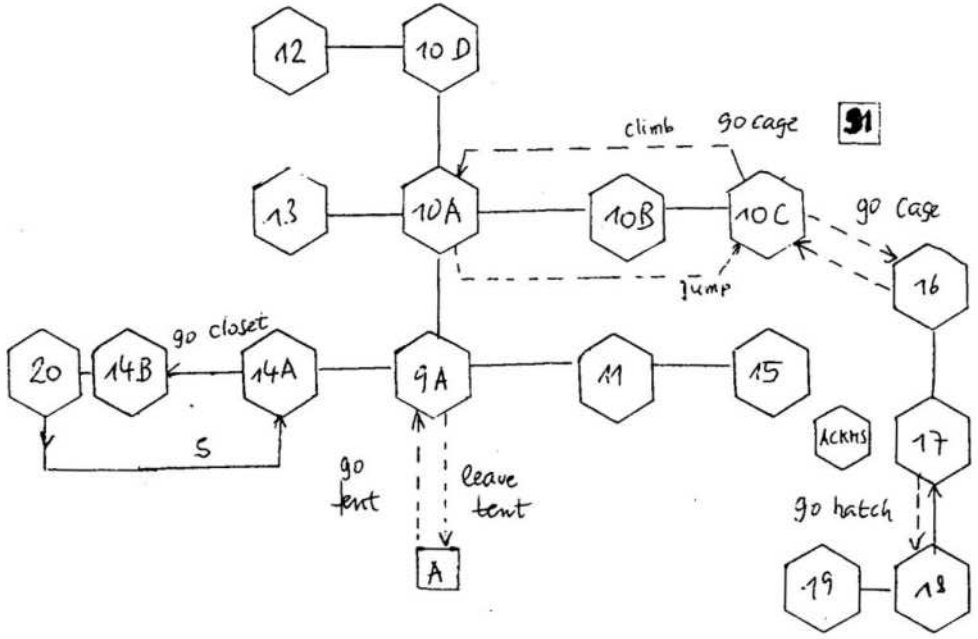
P.S.: An dieser Stelle noch einen herzlichen Dank an Walter Sperl für die freundliche und fundierte Hilfe bei DTP-Problemen!!

```
  10 REM Zeichensatz des Spectrums auf den Drucker uebertragen
  20 REM ===================================================
1000 CLOSE #3
     CLOSE #4
1010 OPEN #3;"t"
     OPEN #4;"b"
     REM Kanaele oeffnen
1020 PRINT #4;CHR$ 27;CHR$ 82;CHR$ 2
     REM Deutscher Zeichensatz an
1030 PRINT #4;CHR$ 27;CHR$ 58;CHR$ 0;CHR$ 0;CHR$ 0
     REM Standard Zeichensatz in's RAM
1040 FOR i=1 TO 10
1050  PRINT #4;CHR$ 27;CHR$ 38;CHR$ 0;
      REM Downlaod character uebertragen (1)
1060  READ a
1070  PRINT #4;CHR$ a;CHR$ a;
      REM Download character uebertragen (2);
          Code des zu uebertragenden Zeichens
1080  FOR k=0 TO 11
1090   READ b
1100   PRINT #4;CHR$ b;
       REM Download character uebertragen (3);
           12 Zahlen pro Zeichen
1110  NEXT k
1120 NEXT i
1130 STOP
2000 REM ===================================================
2010 REM *** DEMO ***
2020 REM Druckerzeichen (deutsch)
2030 FOR i=32 TO 126
2040  PRINT #4;CHR$ i;
2050 NEXT i
2060 PRINT #4;CHR$ 10
2070 REM Druckerzeichen (Sinclair)
2080 PRINT #4;CHR$ 27;CHR$ 37;CHR$ 49;CHR$ 0
     REM Zeichensatz im RAM benutzen
2090 FOR i=32 TO 126
2100  PRINT #4;CHR$ i;
2110 NEXT i
2120 PRINT #4
2130 STOP
2140 REM ===================================================
3000 REM DATA 's fuer die Zeichen
3010 DATA 64,139,124,0,130,16,170,0,170,0,122,0,0
3020 DATA 91,139,0,0,254,0,130,0,130,0,0,0,0
3030 DATA 92,139,0,128,64,32,16,8,4,2,0,0,0
3040 DATA 93,139,0,0,130,0,130,0,254,0,0,0,0
3050 DATA 94,139,32,0,64,0,255,0,64,0,32,0,0
3060 DATA 96,139,18,0,126,128,18,128,2,128,66,0,0
3070 DATA 123,139,0,0,16,0,108,130,0,130,0,0,0
3080 DATA 124,139,0,0,0,0,254,0,0,0,0,0,0
3090 DATA 125,139,0,130,0,130,108,0,16,0,0,0,0
3100 DATA 126,139,64,128,0,128,64,32,0,32,64,0,0
3110 DATA 127,139,124,130,16,130,40,130,40,130,0,130,124
```

CIRCUS !

Locations:

#1	in a field, front of tent	#11	small room
#2	in a field, road	#12	circus ring with cannon
#3	maintainance wagon	#13	mess-room
#4	small enclosure	#14	store-room
#5	field	#15	large tank
#6	by the roadside	#16	gloomy passage
#7	layby, by my car	#17	equipment room
#8	rear of my car	#18	behind a panel
#8a	inside my car	#19	alcove
#9	tent entrance	#20	closet
#10	in the ring		

Procedure:

```
# 5    get shovel, dig, get all
# 8    open boot, get all
# 1    light flashlight, go tent
#20    get whip
#14a   kick chest, get slippers, wear slippers
#10a   inspect rope
#13    get fish
#15    feed sea-lion
#14b   get costume, wear costume, look clown
#10a   get net
#10c   erect net
#10b   climb ladder, go tightrope, north, get bar, jump
# 1    unlight flashlight
# 3    open door, open locker, exam locker, get hacksaw
# 1    light flashlight
#10b   climb ladder, cut tightrope, get cable, jump
# 4    exam generator, repair generator, get handle, start
       generator, siphon petrol
# 8    fill tank
#10a   climb ladder, go trapeze, swing trapeze, cut canvas, jump
#10c   crack whip
#17    look panel, press button
#18    get blueprint, read blueprint, short terminal
#12    get helmet, wear helmet, enter canon, pull lever
# 7    enter car, drive car.
       CONGRATULATIONAS !!!
```